Final Project Report: Real Time Tennis Match Prediction Using Machine Learning

Yang "Eddie" Chen, Yubo Tian, Yi Zhong

December 12, 2017

Our abstract goes here... TO DO

1 Introduction

In the 2017 Stuttgart Open, Roger Federer, then an 18-time grand slam champion, lost to the world No. 302 player Tommy Haas on a grass court, which hadn't happened since 2002. Based on past performance alone, any model would have predicted a Federer win in' pre-game bets. This motivates us to apply machine learning to predict tennis matches in real time; in particular, we want to apply a hybrid model that combines historical and real-time information.

2 Related Work

Research has been done on tennis as it is an ideal candidate for using a hierarchical probability model as a match consists of a sequence of sets, which in turn consist of a sequence of games, which in turn consist of a sequence of points. Recent researchers began to apply machine learning on tennis pre-game predictions based on past performance (including previous encounter, current ranking, etc). Utilizing existing data and building on top of existing research such as Sipko [**?**], Clarke and Dyte [**?**] and Knottenbelt [**?**], our project seeks to apply classification algorithms from machine learning to model men's professional singles matches by using both pre-game, historical performance calcualted via a common opponent model [**?**], and real-time, in-game stats [**?**] [**?**]. As pointed out by Sipko [**?**], this in-game approach "allows the model to capture the change in a player's performance over the course of the match. For example, different players fatigue during a match in different ways." We hope that results from our predictions can be extended to give real-time coaching advice to support game strategy decision.

3 Dataset and Features

3.1 Dataset

We are using two datasets, [**?**] and [**?**]. Both datasets are crowdsourced GitHub repository maintained by Jeff Sackmann. The Match Charting Project contains set level data from 1442 matches, spanning from 1974 to 2017. Tennis ATP contains match level data from 1969 to 2017.

- **Date**: Matches from 2000 ? 2017 are used as samples (1415 matches)
- **Remove duplicates via match id**: Since both datasets are manually entered via crowdsourcing, there are some inaccuracy and duplication.
- **Remove Davis Cup**: Davis Cup is the World Cup" or tennis, where players play for their country. We decided to remove all Davis Cup entry, since group matches may include strategic move that is different from individual tournaments.
- **Merge Tennis ATP and Match Charting Project**: prediction is made on Match Charting Project, where as historical performance for both players is computed from

Tennis ATP. Two datasets are merged on player?s first name and last name, with players with the same names removed manually.

- **Feature Standardization** We are unable to extract all features listed in Sipko [**?**]. We can extract some features from Tennis ATP but not Match Charting Project. We will attempt to extract more features from processing the point by point data in Match Charting Project, and standardize between the historical and current performance.

3.2 Features

We present the list of extracted features in Table **??** for the match-level data, and Table **??** for the set-level data. All features are difference variables unless otherwise noted.

3.2.1 Features from historical match performance

Features are the difference in average for both players with common opponent model.

- **Performance** Number of aces, double faults, break points faced and saved; Percentage of winning on first and second serve, first serve in
- **Match Details** Match duration, number of serve games, number of serve points per game
- **Player Details** ATP Rank, ATP Rank Points, same handedness, height

3.2.2 Features from current match performance

Features are the difference in average for a snapshot in the match: after set 1 and after set 2.

- **Performance** Number of aces, double faults, winners and total points won; Percentage of winning on first and second serve, return points won, first serve in
- **Match Details** Surface (hard, grass, or clay), match type
- **Player Details** Same handedness

Feature	Description
SAME_HANDEDNESS	1 for same handed, 0 for different handed
FSP	First serve success percentage
ACE	Sum of aces
DF	Sum of double faults
WIN	Number of winners
W1SP	Winning on first serve percentage
W2SP	Winning on second serve percentage
RCV	Percentage of return points won
TTL	Total points won
SURFACE	Dummy variable for three types of court
GS	Dummy variable for if tournament is Grand Slam

Table 3.1: Summary of extracted features for previous set performance

3.3 Feature Extraction

Original dataset we used denotes Player 1 as the winner and Player 2 as the loser. While processing, we decide to represent each set in each match with two rows, one from each player's perspective, and label the match result accordingly. Thus we have one row per player per set per match, containing features from historical data and real time data. Historical data includes match details and player's past performance; Real-time data, for the scope of this project, will be defined as the performance from all previous sets in the match:

1. A vector of input features ($\mathbf{X}$), describing the performance in previous set and historical performance from one player's perspective
2. A target value ($\mathbf{y}$), indicating the match outcome

3.3.1 Symmetric Feature Representation

There are two approaches to represent features from two players. First, we can have two features

for the same metric ($RANK_1$ and $RANK_2$). This approach doubles the number of features, but also provides a quantitative measure that we can compare across all players. Second, we can represent difference between two players ($RANK_{DIFF} = RANK_1 - RANK_2$). This would reduce the number of features, but does not allow the model to compare across players. Previous research has shown that the difference variables are predictive enough [?] [?] for past performance (match level). For current performance (set level), we evaluated the performance of both approaches. There was no difference in terms of accuracy, hence the difference model was used for current match as well for consistency.

3.3.2 Common Opponent Model for Past Performance

We use a method proposed by Knottenbelt [?] and extended by Sipko [?] to construct difference variables capturing player characteristics and past performance, by using their **common opponents** to achieve a fair comparison.

- Identify a set of common opponents between the two players of a given match.
- Compute the average performance of both players against the common opponent.
- Take the difference between average performance for each extracted feature.

3.3.3 Edge Cases for Common Opponents

We have run into 2 edge cases: (1) when at least one of the players is new; (2) when two players do not have any common opponents. In our initial investigation, our solution is: (1) when the player is new with no historical data, we remove the samples, which is 84 samples out of 1421 matches; (2) when two players do not have any common opponents, we compute the average for each player (regardless of opponent) and take the difference, which is 74 out of 1421 matches. Note, as described in 2.1.Feature Extraction section, our final sample size will contain 2x * 1421 rows, where x is the number of sets played in each match.

4 Method

4.1 Logistic Regression

Description

4.2 Support Vector Classification

Different Kernel - Linear, RBF, poly

4.3 Other algorithms

TODO Describe your learning algorithms, proposed algorithm(s), or theoretical proof(s). Make sure to include relevant mathematical notation. For example, you can briefly include the SVM optimization objective/formula or say what the softmax function is. It is okay to use formulas from the lecture notes. For each algorithm, give a short description (about 1 paragraph) of how it works. Again, we are looking for your understanding of how these machine learning algorithms work.

5 Experiments, Results, Discussion

As the team is still working on connecting the historical, match-level dataset with the real-time, set-level data sets, we ran some preliminary models fitting the two datasets separately. We use scikit-learn's implementation of the various models [?].

5.1 Historical, Match-level Dataset: Logistic Regression Model

Applying a train-test split of 2:1, we trained a logistic regression model on $\frac{2}{3}$ of all the ATP matches from 2012. Following the labeling from Section **??**, we set $label = 1$ for all the matches with Player 1 being the Winner (notation-only), and then duplicated the entire dataset but swapped Player 1 and Player 2 and set $label = 0$, as well as inverted all difference features.

We achieved an impressive accuracy rate of 87.8% just using the historical, match-level dataset with common opponent model and difference variables.

5.2 Historical, Match-level Dataset: Naive Bayes Classifier

We also ran the data through a Gaussian-based Naive Bayes classifier, which showed an impressive 86.9% accuracy rate (misclassifying 718 points out of 5496). As expected, it was also extremely fast. The impressive performance indicates that the extracted features might indeed be conditionally independent of each other, which is something we never thought of before, and look to explore in the next steps. On a first glance, it seems to suggest

that given Player i won the match, his/her performance on aces and rank point difference or double faults are indeed almost independent. We were surprised by the results, and more sanity check is needed before we put ourselves behind this.

5.3 Real-time, Set-level Dataset: Logistic Regression Model

Additional analysis were done on real-time, set-level data using features listed in Table **??**. We separate our data into two groups: (1) features calculated from performance after 1st set, (2) features calculated from performance after 2nd set. Our hypothesis is that parameters trained on the second group has higher accuracy, since more information regarding the match is known. Our model support this hypothesis. Accuracy on test set for (1) is 77.8%, for (2) is 83.9%. The difference in accuracy between training and testing set is less than 1%, which suggests we do not have an under-fitting problem.

5.3.1 Error Analysis

Same error analysis was done on the logistic model. We found that the most important features are: **W2SP** and **FSP**.

5.4 Real-time, Set-level Dataset: SVM

In addition, we ran the data through a SVM model for the same comparison as logistic regression analysis. We found that while train accuracy increased by around 1%, test accuracy are about 10% lower for (1) and (2). This suggest for SVM may be overfitting the dataset, we will need to do **feature selection** for SVM.

6 Conclusion, Future Work

The use of common opponent model and difference variables already offer high predictive power in simple machine learning models such as logistic regression and Naive Bayes, and it's exciting to see with fine-tuning and more exploration and connecting datasets what we can achieve. It's also helpful to learn that ATP rank point difference between two players alone is already predictive of their likely match outcome.

7 Appendix

Real appendix goes here, takes page count

8 Contributions

8.1 Yang "Eddie" Chen

- Researched and reviewed relevant literature for applicable ideas, such as Clarke and Dyte [**?**] and O'Malley [**?**] for difference variable, and Knottenbelt [**?**] for the common opponent model
- Implemented the Common Opponent Model to look up players stats from common opponents
- Generated ATP Match results for labels
- Typsetted (LaTeX) the project milestone report

8.2 Yubo Tian

- Feature extraction: generated real-time set-level performance data from Sackmann's MCP [**?**]
- Data labeling: extracted match/set results from match data from Sackmann's MCP [**?**]
- Data processing: Joined match-related information, historical data and real-time match data (with Yi); fixed various issues encountered during the process.
- Initial modeling/ Error analyses: Logistic regression on final joined data.

8.3 Yi Zhong

- Feature extraction: get match details related features from MatchChartingProject,
- Data processing: Joined data by linking real-time match data from MCP [**?**] to match details data from ATPMatches [**?**] (methods: playerID, playerName etc) (with Yubo).
- Initial modeling: Logistic regression + SVM with current match data (after set 1 and after set 2)
- Error analyses in the logistic regression and SVM models

References

[1] Stephen R. Clarke and David Dyte. Using official ratings to simulate major tennis tournaments. *International Transactions in Operational Research*, 7(6):585 – 594, 2010.

[2] O'Malley A. James. Probability formulas and statistical analysis in tennis. *Journal of Quantitative Analysis in Sports*, 4(2):1–23, April 2008.

[3] et al. Jeff Sackmann. The tennis abstract match charting project, 2017.

[4] William J. Knottenbelt, Demetris Spanias, and Agnieszka M. Madurska. A common-opponent stochastic model for predicting the outcome of professional tennis matches. *Computers and Mathematics with Applications*, 64(12):3820 – 3827, 2012. Theory and Practice of Stochastic Modeling.

[5] Agnieszka M. Madurska. A set-by-set analysis method for predicting the outcome of professional singles tennis matches. MEng computing- final year project, Imperial College London, amm208@doc.ic.ac.uk, June 2012.

[6] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *ournal of Machine Learning Research*, 12:2825–2830, 2011.

[7] Jeff Sackmann. Atp tennis rankings, results, and stats, 2017.

[8] Michal Sipko. Machine learning for the prediction of professional tennis matches. MEng computing - final year project, Imperial College London, June 2015.